Problem 1.1. The pseudocode is follows.

```
1: procedure SelectionSort(A)
2:     for i = 1 to n − 1 do
3:         min := ∞, minIndex := 0
4:         for j = i to n do
5:             if A[j] < min then
6:                 min = A[j], minIndex = j
7:             end if
8:         end for
9:         A[i], A[j] = A[j], A[i]
10:    end for
11:    return A
12: end procedure
```

Problem 1.2. $\forall_{x,y}\ 1 \leq x < y < i \Rightarrow A[x] \leq A[y]$

Problem 1.3. When $i = n$, $i = j = n$ so swapping the last element with itself is done by line 9, which is totally useless.

Problem 1.4. Both complexities are $\Theta(n^2)$
Best Case: Run statement 6 only once (i.e already sorted)
Worst Case: Run statement 6 for all time (i.e reverse sorted)

Problem 2.1. The proof is follows.
(1) For an array which its length is k, insertion sort runs in $\Theta(k^2)$ in worst-case.
(2) Repeat it for n/k arrays takes $n/k \cdot \Theta(k^2) = \Theta(n/k \cdot k^2) = \Theta(nk)$ in worst-case.

Problem 2.2. The proof is follows.
(1) At first, merge n/k arrays into $n/2k$ arrays. It takes $n/2k \times \Theta(2k) = \Theta(n)$
(2) Merge merged $n/2k$ arrays into $n/4k$ arrays. It takes $n/4k \times \Theta(4k) = \Theta(n)$
(3) Repeat until the arrays merged into 1 arrays.
(4) Each step takes $\Theta(n)$ and the number of steps will be $\lg(n/k)$
(5) $\therefore$ It takes $\Theta(n \lg(n/k))$

Problem 2.3. $k = \Theta(\lg n)$
(1) $\exists n_0, c_1, c_2$ s.t $\forall n \geq n_0\ c_1(n \lg n) \geq c_2(nk + n \lg(n/k)$
(2) $\therefore k \leq c \cdot \lg n$ for some constant c

Problem 2.4. If the array is nearly sorted, insertion sort runs nearly linear, so make k as small enough. Otherwise, make k as large ($< \lg n$) to use merge sort.

Problem 3. The orders are follows.
$2^{n!} > n^n + \ln n = n^n > n! > 10^n + n^{20} > 4^n > e^n > (\lg n)! > 5^{\lg n} = n^{5/2} > n^{2+\sin n} = 5n^2 + 7n > \lg(n!) = n \ln n > 8n + 12 > \sqrt{n} > (\lg n)^2 > \ln(\ln n) > \lg(\lg^* n) > n^{1/\lg n}$

Problem 4.1. True.
(1) $f(n) = O(g(n))$
(2) $\exists n_0, c > 0$ s.t $\forall n \geq n_0$ $f(n) \leq c \cdot g(n)$
(3) $\exists n_0, c > 0$ s.t $\forall n \geq n_0$ $\lg(f(n)) \leq lg(c \cdot g(n)) = \lg c + \lg(g(n)) \leq (1 + \lg c) \lg(g(n))$
(3.1) $\because$ both terms are positive for all sufficiently large n
(4) Then, let $c' = 1 + \lg c$. By definition, $\lg(f(n)) = O(\lg(g(n))$

Problem 4.2. False.
(1) Let $f(n) = 2n$ and $g(n) = n$
(2) Then $2^{f(n)} = 2^{2n} = 4^n$ while $2^{g(n)} = 2^n$

Problem 4.3. False.
(1) Let $f(n) = 1/n$
(2) Then $f(n) = 1/n$ while $(f(n))^2 = 1/n^2$

Problem 4.4. True.
(1) $f(n) = O(g(n))$
(2) $\exists n_0, c > 0$ s.t $\forall n \geq n_0$ $f(n) \leq c \cdot g(n)$
(3) $\exists n_0, > 0$ s.t $\forall n \geq n_0$ $g(n) \geq 1/c_1 \cdot f(n)$
(4) $g(n) = \Omega(f(n))$

Problem 4.5. True.
(1) Let $g(n) = o(f(n))$. Then $\exists n_0, c > 0$ s.t $\forall n \leq n_0$ $g(n) < c \cdot f(n)$
(2) $f(n) + o(f(n)) = f(n) + g(n) < (c + 1) \cdot f(n)$, and $f(n) + o(f(n)) > f(n)$
(3) $\therefore \exists n_0, c > 0$ s.t $\forall n \leq n_0$ $f(n) < f(n) + o(f(n)) < (c + 1) \cdot f(n)$

Problem 5.1. $T(n) = \Theta(n \lg n)$
(1) Since $a = 2, b = 3, f(n) = n \lg n$, $f(n) = \Omega(n^{log_3 2 + \epsilon})$
(2) For sufficiently large n, $af(n/b) = 2(n/3)\lg(n/3) \leq (2/3)n \lg n = cf(n)$ where $c < 1$
(3) Thus, applying master theorem for case 3 get result $T(n) = \Theta(n \lg n)$

Problem 5.2. $T(n) = \Theta(n \lg n)$
(1) Prove that $T(n) \leq cn \lg n$ for some positive constant c
(1.1) Use strong induction on n
(1.2) $T(n) = T(n-2) + \lg n \leq c(n-2)\lg(n-2) + \lg n \leq c(n-1)\lg n \leq cn \lg n$ for $c > 1$
(2) Prove that $T(n) \geq cn \lg n$ for some positive constant c
(2.1) When $n \leq 2$, consider $T(n) = 1$
(2.2) $T(n) = T(n-2) + \lg n = T(n-4) + \lg(n-2) + \lg n = \cdots = 1 + \lg(n!!)$
(2.3) Since $\lg(n!!) > \lg((n/2)!) > n/4 \times \lg(n/4)$, $T(n) = \Omega(n \lg n)$
(3) By (1) and (2), $T(n) = \Theta(n \lg n)$

Problem 5.3. $T(n) = \Theta(n)$
(1) Prove that $T(n) \geq c \cdot n$ for some positive constant c
(1.1) Trivially holds since $T(n) = T(n/2) + T(\sqrt{n}) + n \geq n$
(2) Prove that $T(n) \leq c \cdot n$ for some positive constant c
(2.1) Let $c = 4$ and use strong induction on n

(2.2) When $n \leq 16$, consider $T(n) = 1$. Then, $\forall_{n \leq 4}\ T(n) \leq 4n$
(2.3) Suppose $\forall_{k<n}\ T(k) \leq 4k$
(2.4) Then, $T(n) = T(n/2) + T(\sqrt{n}) + n \leq T(n/2) + T(n/4) + n \leq 2n + n + n = 4n$
(3) By (1) and (2), $T(n) = \Theta(n)$